

Matrikelnummer:

Name:

# Datenbanken 1 (43052)

## 17. Februar 2023

### Hinweise:

- Überprüfen Sie bitte die Klausur auf Vollständigkeit.
- Bitte tragen Sie auf allen Blättern Ihren Namen und Ihre Matrikelnummer ein.
- Zur Bearbeitung der Aufgaben dürfen **keine** Hilfsmittel verwendet werden.
- Machen Sie bitte einen entsprechenden Hinweis, wenn Sie Ihre Lösung nicht unmittelbar unter oder neben der Aufgabenstellung schreiben. Streichen Sie diejenigen Teile der von Ihnen geschriebenen Texte deutlich durch, die nicht in die Bewertung eingehen sollen. Von mehreren angegebenen Lösungen ist die zu bewertende Lösung eindeutig zu kennzeichnen.
- Bei Ihren Lösungen werden außer der korrekten Problemlösung auch die verständliche Formulierung, ausreichende Dokumentation und eine übersichtliche Schreibweise bewertet.
- Bitte schreiben Sie deutlich.

**Viel Erfolg!**

Aufgaben	Erreichbar	Erreicht
1	10	
2	9	
3	25	
4	10	
5	26	
Summe	80	
Prozent	80 %	
SL und Bonus	26 %	
Summe	100 %	

**Aufgabe 1:****10 Punkte**

- (a) Welcher Ebene der 3-Ebenen Schemaarchitektur nach ANSI/SPARCS 1978 ist die folgende Änderung zuzuordnen? (1)

*In MS Access wird ein Eingabeformular zu einer Datenbanktabelle angelegt.*

- (b) Erläutern Sie bitte kurz, was unter einer Transaktion verstanden wird und geben Sie ein Beispiel an. (4)

- (c) Erläutern Sie bitte, welche Wirkungen die folgenden Befehle auf die beiden neuen Nutzer C##Anton und C##Julius haben. (5)

Der Oracle-Systemadministrator führt aus:

```
GRANT DELETE ON KundeTabelle TO C##Anton;  
GRANT SELECT ANY DICTIONARY TO C##Julius;
```

Der Nutzer C##Anton führt danach aus:

```
GRANT DELETE ON KundeTabelle TO C##Julius;
```

Der Systemadministrator führt anschließend aus:

```
REVOKE ALL PRIVILEGES FROM C##Anton;
```

**Aufgabe 2:****9 Punkte**

Betrachten Sie die Relationen  
Weinkeller(Regal, Region, *WeinId*, Flaschen)

mit der folgenden relationalen Anfrage:

$\pi$  Regal, MIN  $\gamma$  Regal, Flaschen;  
MIN(Flaschen) $\rightarrow$ MIN  $\sigma$  Flaschen < 10 Weinkeller  $\cap$   
 $\pi$  Regal, AVG  $\sigma$  Flaschen < 100  
 $\gamma$  Regal, Flaschen; AVG(Flaschen) $\rightarrow$ AVG Weinkeller

Wie lautet die entsprechende SQL-Anfrage?

**Aufgabe 3:****25 Punkte**

- (a) Betrachten Sie die Relation: (5)  
Pflanzenkategorie (Pflanzenkategoriekennziffer, Eigenschaft, *Pflanzenoberkategorie*)  
Das optionale Fremdschlüsselattribut *Pflanzenoberkategorie* referenziert die Relation Pflanzenkategorie.  
Es soll die Eigenschaft der Pflanzenkategorie zusammen mit der Eigenschaft der zugehörigen Pflanzenoberkategorie gelistet werden.  
Die Anfrage soll nur einen Verbundoperator enthalten und keine WHERE-Bedingung verwenden. Wie lautet die zugehörige SQL-Anfrage?

- (b) Betrachten Sie die Relationen: (10)  
Studiengang(StudiengangId, Matrikelnummer, Status, Einschreibedatum)  
Leistung(Modulnummer, *StudiengangId*, *VeranstaltungsId*, ECTSPunkte)

Listen Sie die Studiengänge mit dem Status *rosa*, die Leistungen mit einem ECTS-Punkt größer als 10 beinhalten.

Geben Sie die zugehörigen StudiengangIds und Matrikelnummern absteigend sortiert an.

Verwenden Sie zur Lösung **Unterabfragen** und **Mengenquantoren**, jedoch keine Verbundoperatoren, Negationen oder Mengenoperatoren.

- (c) Betrachten Sie die Relation Studierende (Matrikelnummer, Vorname, Nachname, Abschluss, Studiengang, Immatrikulationsdatum) (10)

Listen Sie die Anzahl der Studierenden mit einem Immatrikulationsdatum im Monat Dezember und einem Nachnamen mit mehr als 6 Buchstaben pro Studiengang und Abschluss.

Die Anzahl der Studierenden soll nur ausgegeben werden, wenn diese größer als 10 ist. Die Ausgabe soll nach der Anzahl absteigend sortiert werden.

Verwenden Sie zur Lösung keine Unterabfragen und keine Mengenoperatoren.

**Aufgabe 4:****10 Punkte**

Betrachten Sie die Relationen und die zugehörige Benutzersicht.

Dozent(DozentId, Titel, Vorname, Nachname, Fachbereich)

Veranstaltung(VeranstaltungsId, *DozentId*, Status, Prüfungsdatum)

```
CREATE VIEW DozentView
```

```
AS
```

```
SELECT DozentId, Nachname, Vorname FROM Dozent k
```

```
WHERE 0 < (SELECT Count(*) FROM Veranstaltung b WHERE b.DozentId=k.DozentId)
```

```
WITH CHECK OPTION
```

a) Erläutern Sie bitte, welche Daten die Benutzersicht anzeigt.

b) Was soll die folgende SQL-Anweisung machen?

```
INSERT INTO DozentView VALUES(2310, 'xyz', 'abc');
```

c) Wird diese SQL-Anweisung ausgeführt? Begründen Sie Ihre Entscheidung.



**Aufgabe 5:****26 Punkte**

- (a) Das folgende Datenbankprogramm ist fehlerhaft. Markieren und erläutern Sie die vorhandenen Fehler. (8)

```
CREATE OR REPLACE STORED FUNCTION mytest (var IN Artikel.Preis%TYPE)
RETURN NUMERIC
IS BEGIN
DELETE FROM Kunde;
RAISE EXCEPTION ('FEHLER');
END;
```

- (b) Betrachten Sie die folgende Relation: (18)  
Bestellung (Bestellnummer, *Kundennummer*, Bestellstatus, Bestelldatum)

Wenn eine neue Bestellung eines Kunden eingeht, dann soll automatisch geprüft werden, ob mindestens drei weitere Bestellungen des Kunden innerhalb der letzten 14 Tage vorliegen. Wenn dies der Fall ist, dann wird der Bestellstatus auf *bestellt(VIP)* gesetzt. Wenn mindestens eine andere Bestellung vorliegt, dann wird der Bestellstatus auf *bestellt* gesetzt, ansonsten auf *prüfen*.

Setzen Sie diese Integritätsregel in Oracle um und erläutern Sie unter Angabe von SQL-Befehlen, wie die Korrektheit der Implementierung getestet werden kann. Achten Sie auf eine strukturierte Programmierung.